Creating and Using Plugins in the Swing SDK

EditLive! for Java Swing's plugin functionality is only supported with an EditLive! for Java Swing Enterprise Edition license.

EditLive! for Java Swing allows developers to easily create and integrate plugins into the editor. Plugins deliver additional functionality via Java classes that adhere to the specifications listed below.

Creating Plugins

Plugins are defined using the Plugin XML specifications. Plugin XML tells the editor meta information such as:

- The location of the plugin code.
- Changes to the menus in the editor to allow the user to utilize new functionality provided in the plugin.
- Definitions on when the plugin should be loaded.

To create an EditLive! for Java Swing plugin, you need to perform the following steps:

1. Create the desired functionality via Java classes as specified below.
2. Create Plugin XML to reference the functionality you created in step 1.
3. If required, create menu items in the Plugin XML to allow users to enable and utilize the functionality in your plugin.

Coding Java Classes to be Used as Plugins

When creating Java code to be loaded as an EditLive! for Java Swing plugin, the following steps need to be taken:

1. Your compiled Java classes need to be stored in a jar archive. This archive will be referenced via the <advancedapis (Applet)> Plugin XML.
2. The central class for your plugin (i.e. the Java class defined in the class attribute for the <advancedapis (Applet)> element) needs a constructor which accepts a single parameter: an instance of ELJBean.

Loading Plugins

EditLive! for Java loads plugins by registering Plugin XML with the editor. Plugin XML can be registered with the editor via the following methods:

- <pluginss> Configuration File Element